

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

Please cancel Claims 1-21.

22. (New) A method of accessing an entry in a list of entries in a computer system, each entry in the list of entries including a next entry pointer that points to another entry in the list such that the next entry pointers together form a closed loop, the method comprising:

statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry;

further determining whether the access probability has a predetermined probability magnitude;

reading a start pointer pointing to one of the entries in the list;

examining the entries in the list in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found;

accessing the statistically determined entry;

overwriting the start pointer in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically determined entry when the access probability has the predetermined probability magnitude; and

following the overwritten pointer to the statistically determined entry during the future access to the list so as to make the statistically determined entry the first entry accessed during the future access to the list.

23. (New) A method as claimed in claim 22, further comprising terminating examination of the entries in the event that the statistically determined entry is not found and all of the entries in the list have been examined.
24. (New) A method as claimed in claim 22, wherein accessing the statistically determined entry includes reading the entry.
25. (New) A method as claimed in claim 22, wherein accessing the statistically determined entry includes writing to the entry and the method further comprises imposing a mutex.
26. (New) A method as claimed in claim 22, wherein overwriting the start pointer does not require a mutex.
27. (New) A method as claimed in claim 22, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.

28. (New) A method of accessing an entry in a list of entries in a computer system, each of the entries including a next entry pointer that points to another entry in the list such that the next entry pointers together form a closed loop, the method comprising:

statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry;

further determining whether the access probability has a predetermined magnitude;

reading a start pointer pointing to one of the entries in the list;

examining the entries in the list in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found;

if the statistically determined entry is found then performing the following steps:

accessing the statistically determined entry;

overwriting the start pointer in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically determined entry when the access probability has the predetermined magnitude; and

if the statistically determined entry is not found and all of the entries in the list have been examined then performing the following step:

terminating examination of the entries.

29. (New) A method as claimed in claim 28, wherein accessing the statistically determined entry includes reading the entry.
30. (New) A method as claimed in claim 28, wherein accessing the statistically determined entry includes writing to the entry and the method further comprises imposing a mutex.
31. (New) A method as claimed in claim 28, wherein overwriting the start pointer does not require a mutex.
32. (New) A method as claimed in claim 28, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.
33. (New) A method of accessing an entry in a list of entries in a computer system, each of the entries including a next entry pointer that refers to another entry in the list such that the next entry pointers together form a closed loop, the method comprising:

statistically determining an access probability that a future access of the list will require a last-accessed entry in the list in order to provide a statistically determined entry;

further determining whether the access probability has a predetermined probability magnitude;

reading a start pointer pointing to one of the entries in the list;

examining the entries in the list in turn beginning with the entry pointed to by the start pointer and continuing until the statistically determined entry is found;

accessing the statistically determined entry; and

overwriting the start pointer in accordance with the further determining in order to provide an overwritten pointer so as to point to the statistically determined entry when the access probability has the predetermined probability magnitude; and

following the overwritten pointer to the statistically determined entry during the future access to the list so as to make the statistically determined entry the first entry accessed during the future access to the list.

34. (New) A method as claimed in claim 33, further comprising terminating examination of the entries in the event that the statistically determined entry is not found and all of the entries in the list have been examined.
35. (New) A method as claimed in claim 33, wherein the accessing and overwriting steps are omitted when the statistically determined entry is not found.
36. (New) A method as claimed in claim 33, wherein accessing the statistically determined entry includes reading the entry.

37. (New) A method as claimed in claim 33, wherein accessing the statistically determined entry includes writing to the entry and the method further comprises imposing a mutex.
38. (New) A method as claimed in claim 33, wherein overwriting the start pointer does not require a mutex.
39. (New) A method as claimed in claim 33, wherein the computer system is a multi-threaded environment and the overwriting step is performed atomically.